



Bridge AI — API Testing Documentation

Base URL: `http://127.0.0.1:8001/api/v1`

Environment: Local Development

Authentication: JWT (Access Token)

Tested Using: Postman



1. Signup

Endpoint:

`POST /signup`

Description:

Creates a new user account and sends verification email.

Request Body (JSON):

```
{
  "email": "ankit.mishra@atlanticksolutions.com",
  "password": "Pass@123"
}
```

Response:

```
{
  "message": "Data stored successfully, verification email sent."
}
```



Status: 200 OK — Success



2. Email Verification

Endpoint:

`POST /verify-email`

Request Body:

```
{
  "email": "ankit.mishra@atlanticksolutions.com",
  "code": "899296"
}
```

Response:

```
{
  "message": "Email verified successfully"
}
```



Status: 200 OK — Success



3. Get Current Logged-In User

Endpoint:

GET /me

Headers:

Authorization: Bearer <ACCESS_TOKEN>

Response:

```
{
  "id": "cmhkkdmn70000k1i6wpc628to",
  "email": "ankit.mishra@atlanticksolutions.com",
  "name": "Ankit Mishra",
  "phone": "9098821587",
  "subscriptionType": "free",
  "billingCycle": "none",
  "availableCredits": 0
}
```



Status: 200 OK — Success



4. Subscription Details

Endpoint:

GET /subscription

Headers:

Authorization: Bearer <ACCESS_TOKEN>

Response:

```
{
  "subscription_type": "free",
  "billing_cycle": "none",
  "available_credits": 0,
  "last_credit_reset": null
}
```



Status: 200 OK — Success



5. Forgot Password

Endpoint:

POST /forgot-password

Request:

```
{
  "email": "ankit.mishra@atlanticksolutions.com"
}
```

Response:

```
{
  "message": "Password change verification code sent to
ankit.mishra@atlanticksolutions.com"
}
```

✅ **Status:** 200 OK — Success



6. Reset Password

Endpoint:

POST /reset-password

Request:

```
{
  "email": "ankit.mishra@atlanticksolutions.com",
  "verification_code": "451508",
  "new_password": "Ankit@123",
  "confirm_password": "Ankit@123"
}
```

Response:

```
{
  "message": "Password reset successfully"
}
```

✅ **Status:** 200 OK — Success



7. Update Profile

Endpoint:

PUT /update-profile

Headers:

Authorization: Bearer <ACCESS_TOKEN>
Content-Type: multipart/form-data

Form Data:

name: Ankit Mishra
phone: 9098821587

Response:

```
{
  "id": "cmhkkdmn70000k1i6wpc628to",
  "email": "ankit.mishra@atlanticksolutions.com",
  "name": "Ankit Mishra",
  "phone": "9098821587",
  "subscriptionType": "free",
  "billingCycle": "none"
}
```

}

✓ **Status:** 200 OK — Success

8. Refresh Token

Endpoint:

POST /token/refresh

Request:

```
{
  "refresh_token": "<YOUR_REFRESH_TOKEN>"
}
```

Response:

```
{
  "access_token": "<NEW_ACCESS_TOKEN>",
  "token_type": "bearer",
  "user": {
    "email": "ankit.mishra@atlanticksolutions.com",
    "name": "Ankit Mishra"
  }
}
```

✓ **Status:** 200 OK — Success

9. Create New Website Scan

Endpoint:

POST /scan

Request:

```
{
  "url": "https://portfolio-website-flame-xi-26.vercel.app"
}
```

Response:

```
{
  "id": "cmhlntf6600031vtoysgxw9zq",
  "status": "pending"
}
```

✓ **Status:** 200 OK — Success

10. Get Scan Result by ID

Endpoint:

GET /scan/{scan_id}

Example:

GET /scan/cmhlntf6600031vtoysgxw9zq

Response:

```
{
  "id": "cmhlntf6600031vtoysgxw9zq",
  "status": "completed",
  "report": { "pillar_scores": {...}, "recommendations": [...] }
}
```

✅ **Status:** 200 OK — Success

11. Get All Scans

Endpoint:

GET /scans

Response:

List of all scans stored in the database.

✅ **Status:** 200 OK — Success

12. Create Subscription

Endpoint:

POST /create-subscription

Request:

```
{
  "plan_type": "pro",
  "billing_cycle": "yearly"
}
```

Response (example):

```
{
  "message": "Subscription created successfully",
  "razorpay_order_id": "order_LyA1234xyz",
  "plan_type": "pro",
  "billing_cycle": "yearly"
}
```

✅ **Status:** 200 OK — Success

13. Razorpay Webhook (Subscription Activated)

Endpoint:

POST /razorpay-webhook

Headers:

X-Razorpay-Signature: <computed-signature>
Content-Type: application/json

Body:

```
{  
  "event": "subscription.activated",  
  "payload": {  
    "subscription": {  
      "entity": {  
        "id": "sub_1234567890abcdef",  
        "status": "active"  
      }  
    }  
  }  
}
```



Expected Status: 200 OK



If 400: Signature verification failed



The webhook will return:

200 OK – Webhook verified successfully!